SRv6 Operations
Internet-Draft
Intended status: Informational
Expires: 5 January 2026

C. Filsfils
P. Camarillo, Ed.
K. Michielsen
Cisco Systems
A. Gorovoy
Nebius

4 July 2025

SRv6 End-to-End DC Frontend and WAN
draft-filsfils-srv6ops-srv6-e2e-dc-frontend-wan-00

Abstract

The SRv6 Network Programming architecture allows an application to control the end-to-end journey of traffic flows through different network domains in a unified and stateless manner, meaning intermediate network devices do not store per-flow information.

This document covers its application to the integration of data center (DC) and wide area network (WAN) architectures using SRv6 with uSID (NEXT-CSID). This eliminates the complexities and inefficiencies associated with traditional fragmented network designs. The solution enhances scalability and enables flexible stateless service insertion by unifying the DC and WAN under a single SRv6 domain.

Table of Contents

1. Introduction

Traditionally, Data Centers (DCs) and Wide Area Networks (WANs) are designed and operated as independent network domains, each with its own architecture. This separation introduces operational complexity and makes it challenging to seamlessly integrate and deploy network services, such as security appliances or load balancers, across the end-to-end path.

Segment Routing over IPv6 (SRv6) [RFC8986] enables a unified approach to networking by leveraging source routing.

SRv6 uSID (NEXT-CSID), [RFC9800]) provides:

* *Stateless Traffic Engineering*: Any path in the fabric can be enforced by simply encoding a sequence of SRv6 instructions in the packet header. There is *no per-flow state in the fabric* (unlike MPLS RSVP-TE).

* *Stateless Service Insertion*: Any service (e.g., Firewall, IDS) may be inserted in the packet delivery path. This is achieved simply by adding the corresponding segment to the packet. *There is no per-flow state required on the services*.

* *Integrated Overlay*: SRv6 provides per-tenant segmentation, ensuring flows from different tenants are isolated while sharing the same physical infrastructure. This capability may be combined with stateless traffic engineering and stateless service insertion, *without requiring any additional overlay encapsulations*.

* *End-to-end, with minimum overhead*: A simple *IPv6 encapsulation suffices to encode up to six uSIDs in the Destination Address*. These instructions encode a path end-to-end across domains in a provider network. If more instructions are required, a simple Segment Routing extension Header (SRH) [RFC8754] can be added.

This document presents an SRv6 end-to-end solution for the integration of data center (DC) and wide area network (WAN), eliminating the complexity and inefficiencies of traditional fragmented architectures. By extending SRv6 capabilities across both environments, the architecture removes the necessity for traditional, often complex, Data Center Interconnect (DCI) gateways that typically translate between separate DC and WAN routing domains.

SRv6 uSID (NEXT-CSID) removes protocol translation overhead, improves scalability, and enables flexible service insertion (e.g., firewall services).

The document [draft-filsfils-srv6ops-srv6-ai-backend] documents the usage of SRv6 to optimize workloads in the AI backend.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

SRv6 Segment Routing over IPv6 [RFC8986].

uSID Micro-segment Identifier, formally defined as NEXT-CSID in [RFC9800].

The term _uSID (micro SID)_ predates the formal naming and has been widely adopted across the industry - including operators with large-scale deployments, vendors, open-source implementations, and used consistently in multi-vendor interoperability reports.

To maintain alignment with the formal specification while also acknowledging the widespread and practical use of the term, this document uses uSID and NEXT-CSID interchangeably.

ECMP Equal-Cost Multi-Path

uN uSID associated with a Node, the short notation for the End behavior with NEXT-CSID, PSP, and USD flavors as defined in [RFC9800].

uA uSID associated with an Adjacency, the short notation for the End.X behavior with NEXT-CSID, PSP, and USD flavors [RFC9800].

VXLAN Virtual Extensible LAN [RFC7348]

DCI Data Center Interconnect, interconnecting DC and WAN domains

PE Provider Edge router

TOR Top-of-Rack router

BR Border Router, interconnecting WAN domain and the Internet

3. Initial Fragmented Network Architecture

3.1. Overview

The traditional data center architecture utilizes VXLAN for intra-DC communication, while the WAN employs SR-MPLS. Stitching these domains requires complex DCI gateway functions and protocol translations, which increase operational overhead and hinder service agility. These gateways translate between VXLAN and SR-MPLS.

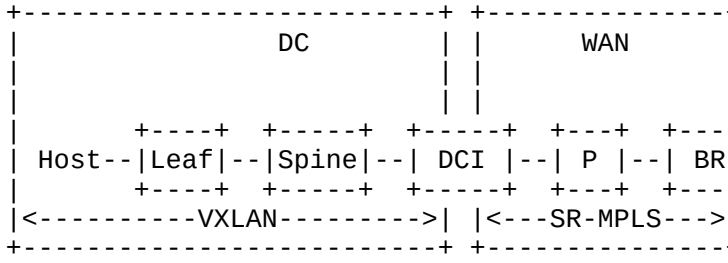

Figure 1: Traditional data center architecture

This is depicted in Figure 1. The Host sends traffic towards the Internet. Upon traffic reception at the DC, it is encapsulated with VXLAN. This packet continues all the way up to the DCI, where there is protocol translation between VXLAN and SR-MPLS, requiring information that needs to be carried from one data-plane to another. Once the packet reaches the Border Router (BR), the SR-MPLS encapsulation is removed, and the packet continues up to the Internet.

3.2. Challenges with a Fragmented DC/WAN

This design presents the following challenges:

* *DCI Protocol Translation*:
  - Performance Penalty: Some hardware incurs in a performance degradation.
  - Availability: Not all vendors support the functionality homogenously.
  - Scalability bottleneck: These boxes maintain VRF state, reducing scalability.
  - Complexity: Information like Group Policy Objects (GPO), tags, or context must be conveyed across and between the DCs.
  - Reliability: The increased complexity introduces additional potential points of failure.
  - Cost: The increased functionality typically incurs an additional cost.

* *Service Insertion*: insertion of services, such as firewalls, require complex policies, with rigid traffic steering.

  This results in inefficient designs. For example, in the case of firewall services, due to the rigid routing, firewalls are typically deployed as a large cluster near the network perimeter. This results in scalability challenges (i.e., stateful firewall state synchronization between instances).

* *IPv4 Loopback dependency*: Many vendors still require IPv4 loopbacks on both VXLAN and SR-MPLS networks, adding unnecessary operational complexity and limiting pure IPv6 deployments.

4. End-to-End SRv6 across DC Frontend and WAN

The transition from a fragmented network architecture to an end-to-end SRv6 uSID design enables seamless communication between data center (DC) workloads and the wide area network (WAN).

The key benefits of this architecture include:

* *Elimination of protocol translation*: Removing VXLAN-to-SR-MPLS protocol conversion improves efficiency and reduces processing overhead at the DCI.

* *Enhanced scalability*: The nature of SRv6 supports flexible traffic engineering with no per-flow state in the fabric.

* *Simplified service chaining*: Native SRv6 capabilities enable the efficient insertion of services without requiring additional protocols or complex policy-based routing policies.

* *Optimized forwarding with minimum MTU*: A simple IPv6 encapsulation allows encoding up to six instructions in the IPv6 Destination Address. If additional instructions are needed, a Segment Routing extension Header can be added to encode additional instructions.

* *IPv6-native*: The architecture enables a fully IPv6-native deployment, eliminating dependencies on IPv4 loopbacks while simplifying management. Transitioning away from fragmented IPv4 dependencies eases uniform addressing across the entire network fabric.

4.1. Service Insertion

Service insertion typically relies on complex policy-based routing mechanisms that direct traffic through predefined service clusters, such as firewalls. These approaches often lead to scalability limitations and suboptimal traffic paths. By using SRv6 services can be seamlessly integrated into the forwarding plane without requiring additional tunneling mechanisms or extensive policy configurations.

Services are inserted into the forwarding path for a specific flow simply by adding the corresponding uSID to the source routed network program. This allows:

* *Stateless steering*: SRv6 uSID (NEXT-CSID) encoding removes the need for per-flow state on intermediate nodes, improving scalability.

* *Flexible service chaining*: Multiple services, such as firewalls, load-balancers, intrusion detection (IDS), or deep packet inspection (DPI), can be chained together dynamically by appending additional uSIDs.

Using the firewall service as an example, when a workload within the DC initiates traffic requiring inspection, the ingress node includes the necessary SIDs in the SRv6 network program. These SIDs guide the packet through the firewall, ensuring compliance with security policies. The firewall processes the traffic and forwards it along the next segment in the chain, with minimal overhead.

5. Illustration

In this section we illustrate how an end-to-end SRv6 converged DC/WAN architecture operates.

5.1. Unsecured Traffic Flow

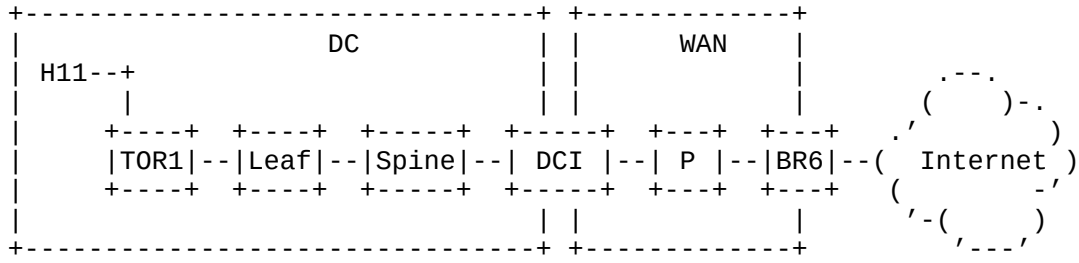

Figure 2: Reference network topology

Figure 2 illustrates one DC and the WAN. The network within the DC has a spine-leaf Clos design. BGP is used as the routing protocol in the DC. The WAN network uses IS-IS as the IGP. Other routing protocols may be used in the DC and the WAN. The Border Router (BR) connects the WAN to the Internet.

SRv6 is enabled on nodes TOR1 and BR6. The remaining nodes are performing IPv6 forwarding, and do not require to be SRv6 enabled.

The operator provisions the following:

* SRv6 SID Space in the fabric 5f00:0::/32

* TOR1 locator prefix: 5f00:0:1::/48. This prefix is advertised in the network.

* BR locator prefix: 5f00:0:6::/48. This prefix is advertised in the network.

Hosts H11 represents a virtual workload connected to a TOR. The TOR and BR act as SRv6 Provider Edge (PE) nodes, providing SRv6 L3VPN connectivity between the hosts and the Internet. The SRv6 PEs advertise their VPN routes in BGP [RFC9252].

As an L3VPN PE, BR6 advertises its IPv4 and IPv6 Internet reachability in its VRF INTERNET to the TORs using BGP. The BR typically advertises its Internet reachability as a default route. A single BR is illustrated, but multiple BRs may exist for redundancy.

As an SRv6 L3VPN PE, BR6 advertises the VPN routes with an SRv6 L3VPN service SID of behavior End.DT4 for IPv4, End.DT6 for IPv6, or End.DT46 for both IPv4 and IPv6 [RFC8986]. BR6 advertises the SRv6 L3VPN service SID 5f00:0:6:e000:: for the Internet routes, with 5f00:0:6::/48 an SRv6 locator of BR6.

TOR1 is configured with a VRF named UNSECURED for unsecured connectivity. Host H11 is connected to TOR1 in VRF UNSECURED. TOR1 imports the L3VPN routes advertised by BR6 into this VRF.

TOR1 advertises the IPv4 and IPv6 service routes in VRF UNSECURED with an SRv6 L3VPN service SID 5f00:0:1:e001::. BR6 imports the routes of this VRF into its single local VRF INTERNET.

TOR1 steers a traffic flow from H11 in VRF UNSECURED to the Internet via BR6. As a result:

* TOR1: encapsulates packets with IPv6 Destination Address: 5f00:0:6:e000::

* Leaf, Spine, DCI, P: perform IPv6 routing based on the prefix 5f00:0:6::/48

* BR6: receives the packet with IPv6 DA 5f00:0:6:e000::. This matches a FIB entry locally instantiated as an SRv6 SID associated with the End.DT behavior. As a result, it decapsulates the packet and forwards the exposed inner packet to the internet.

The reverse direction is symmetrical.

Note that in this illustration, the first SRv6-capable node is the TOR switch, which encapsulates traffic with the appropriate SRv6 policy towards the destination. However, in some deployments, the SRv6 encapsulation may be performed by a virtualized gateway (e.g., a software router or hypervisor-based function), making the TOR, Leaf, and Spine devices operate purely as IPv6 forwarders. This flexibility enables SRv6 to integrate seamlessly into diverse data center designs without requiring SRv6 support across the entire fabric.

5.2. Secured Traffic Flow

This section describes the scenario of an SRv6-unaware [I-D.ietf-spring-sr-service-programming] standalone firewall service. SR Policies [RFC9256] steer the packet flows requiring inspection through the firewall.

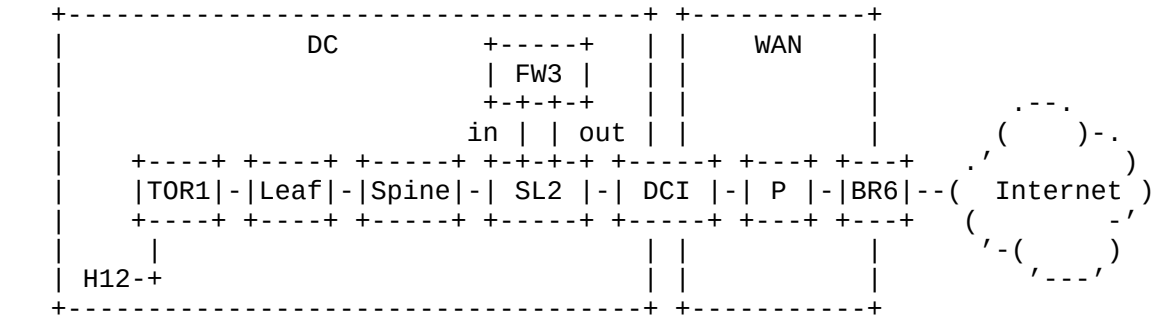

Figure 3: Reference network topology with standalone firewall service

Figure 3 illustrates the network topology with a firewall service FW3. This firewall service is connected to service leaf (SL) node SL2. The firewall FW3 is dual-connected to the SL node via an inside interface (FW3-IN) and outside interface (FW3-OUT).

FW3 inspects the received packets and forwards the allowed ones. FW3 inspects the inner packet and does not modify the outer IPv6 header, apart from the HL field.

The SL node SL2 is SRv6-enabled and advertises a locator prefix 5f00:0:2::/48.

SL2 allocates and installs a uA SID (5f00:0:2:e000::) directing packets to the inside interface of the firewall (FW3-IN) and a uA SID (5f00:0:2:e001::) directing packets to the outside interface of the firewall (FW3-OUT).

Traffic from the Host to the Internet:

For TOR1 to direct service traffic flows through the firewall FW3, it does the following:

* *TOR1*: Encapsulates the traffic from H12 with an IPv6 header with DA 5f00:0:2:e000:6:e000:: .
  - This network program instructs the packet to go to node SL2, take the interface FW3-IN, go to node BR6, execute the VPN behavior.
* *Leaf*:
  - Packet in: DA=5f00:0:2:e000:6:e000::
  - Lookup in FIB, and forward according to the route for prefix 5f00:0:2::/48
  - Packet Out: DA=5f00:0:2:e000:6:e000::
* *Spine*:
  - Packet in: DA=5f00:0:2:e000:6:e000::
  - Lookup in FIB, and forward according to the route for prefix 5f00:0:2::/48
  - Packet Out: DA=5f00:0:2:e000:6:e000::
* *SL2*:
  - Packet in: DA=5f00:0:2:e000:6:e000::
  - Lookup in FIB. Match in FIB entry 5f00:0:2:e000::, instantiated as a local uA SID. Process and xconnect on the interface FW3-in.
  - Packet Out: DA=5f00:0:6:e000::
* *FW3*: Process the inner header of the packet according to security policy.
* *DCI*:
  - Packet in: DA=5f00:0:6:e000::

- Lookup in FIB, and forward according to the route for prefix 5f00:0:6::/48

- Packet Out: DA=5f00:0:6:e000::

* *P*:

- Packet in: DA=5f00:0:6:e000::

- Lookup in FIB, and forward according to the route for prefix 5f00:0:6::/48

- Packet Out: DA=5f00:0:6:e000::

* *BR6*:

- Packet in: DA=5f00:0:6:e000::

- Lookup in FIB. Match in FIB entry 5f00:0:6:e000::, instantiated as a local End.DT SID. Decapsulate and forward the inner packet towards the internet.

The traffic flow from the Internet to the Host is symmetrical, with the remark that on SL2 the SID 5f00:0:2:e001 is used to xconnect the packet on the FW3-out interface.

This configuration ensures that forward and reverse traffic flows pass through the same firewall. The illustration is expanded in Appendix A.1 showing how this use-case is realized from a control-plane point of view leveraging the SR Policy and the notion of Color-Only BGP Destination Steering [RFC9256].

Note that in a typical deployment the service (e.g, Firewall) may be clustered. This is grouping multiple service instances as a single logical device. The connections are synchronized across the instances, which ensures redundancy. SRv6 uSID supports this easily by leveraging an anycast service SID. This is detailed in Appendix A.2.

In Appendix B we remind the difference between SRv6-aware and SRv6-unaware services, and provide considerations for each of them.

6. Benefits

* *Gateway Removal*: Eliminates stateful DCI translation, removing a critical bottleneck in traditional architectures. This leads to better performance, reliability, scalability, and lower operational costs.

* *End-to-End*: Unified underlay and overlay from DC to WAN.

* *Scalability*: The intermediate nodes do not maintain any per-flow state.

* *Lower MTU Tax*: SRv6 uSID (NEXT-CSID) allows encoding up to six uSIDs within the IPv6 Destination Address field. This eliminates the VXLAN overhead, improving transport efficiency.

* *Flexible Service Insertion*: Dynamically steer selected traffic through services (e.g., security appliances) without complex policy-based routing mechanisms. Insert services anywhere in the network and apply the policy on a per-vm, per-prefix, or per-customer basis.

* *Smaller Clusters*: Due to the flexible traffic steering capabilities provided by SRv6, the operator may choose to deploy smaller service clusters (e.g., firewall clusters) distributed along the network. This drastically reduces the scaling challenge of clusters.

* *Egress Peer Engineering (EPE)*: SRv6 enables precise outbound traffic control, optimizing interconnection with external networks.

7. Security Considerations

The deployment model described in this document does not require any new security mechanism aside of those defined in [RFC8986].

8. Acknowledgements

The authors would like to recognize the work of Alexey Gorovoy, Andrew Tikhonov, and Samvel Vartapetov from Nebius.

Alexey Gorovoy presented this use case at MPLS & SRv6 World Congress in March 2025. A recording is available here: https://www.segment-routing.net/conferences/Paris25-Nebius-Alexey-Gorovoy/

9. References

9.1. Normative References

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC9012] Patel, K., Van de Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", RFC 9012, DOI 10.17487/RFC9012, April 2021, <https://www.rfc-editor.org/info/rfc9012>.

[RFC9252] Dawra, G., Ed., Talaulikar, K., Ed., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)", RFC 9252, DOI 10.17487/RFC9252, July 2022, <https://www.rfc-editor.org/info/rfc9252>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

[RFC9800] Cheng, W., Ed., Filsfils, C., Li, Z., Decraene, B., and F. Clad, Ed., "Compressed SRv6 Segment List Encoding", RFC 9800, DOI 10.17487/RFC9800, June 2025, <https://www.rfc-editor.org/info/rfc9800>.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-service-programming-11, 23 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-11>.

[I-D.ietf-idr-sr-policy-safi]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., and D. Jain, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-sr-policy-safi-13, 6 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-sr-policy-safi-13>.

9.2. Informative References

Appendix A. Illustration extension

A.1. Control Plane for Secured Traffic Flow

Figure 3 illustrates the network topology with a firewall service FW3. This firewall service is connected to service leaf (SL) node SL2. The firewall FW3 is dual-connected to the SL node via an inside interface (FW3-IN) and outside interface (FW3-OUT).

FW3 inspects the packets received on FW3-IN and sends the allowed packets to FW3-OUT. FW3 inspects the inner packet and does not modify the outer IPv6 header, apart from the HL field.

The SL node SL2 is SRv6-enabled and advertises a locator prefix 5f00:0:2::/48.

SL2 allocates and installs a uA SID (5f00:0:2:e000::) directing packets to the inside interface of the firewall (FW3-IN) and a uA SID (5f00:0:2:e001::) directing packets to the outside interface of the firewall (FW3-OUT).

For TOR1 to direct service traffic flows through the firewall FW3, TOR1 steers the packets into an SR Policy named POLTOR, with a SID list <5f00:0:2:e000::>. This SID list steers the packets to SL2, directing them into the inside interface FW3-IN of FW3. SR Policy POLTOR has a color COLFIREWALL and an IPv6 null endpoint (::) ([RFC9256]).

For BR6 to direct service traffic flows through the firewall FW3, BR6 steers the packets into an SR Policy named POLBR, with a SID list <5f00:0:2:e001::>. This SID list steers the packets to SL2, directing them into the outside interface FW3-OUT of FW3. SR Policy POLBR has a color COLFW3 and an IPv6 null endpoint (::).

The devices connected to TOR1 that require firewall inspection of their communication with the Internet, such as host H12, are placed in a VRF named SECURED on TOR1.

As an SRv6 L3VPN PE, TOR1 advertises the IPv4 and IPv6 service routes of VRF SECURED with an SRv6 L3VPN service SID 5f00:0:1:e000:: and a Color Extended Community ([RFC9012]) named COLFW3 for ease of reference. COLFW3 has Color-only Types Field ([draft-ietf-idr-sr-policy-safi]) set to 1 (Specific or Null Endpoint Match) to enable the routes with this color for Color-Only BGP Destination Steering ([RFC9256]).

BR6 imports the routes of VRF SECURED into its single local VRF INTERNET. Since the routes have a color COLFW3 of Color-only type 1, and there is an active color-only SR Policy POLBR with color COLFW3 on BR6, BR6 steers the imported routes into SR Policy POLBR.

Consequently, packets arriving at BR6 from the Internet, destined for H12, are encapsulated in an outer IPv6 header with a SID list <5f00:0:2:e001::, 5f00:0:1:e000::>. This SID list may be combined into a single uSID container 5f00:0:2:e001:1:e000:: in the IPv6 DA. This SID list directs the encapsulated packets through FW3 and then to TOR1.

BR6 advertises its VRF INTERNET routes with an SRv6 L3VPN service SID 5f00:0:6:e000:: without a Color Extended Community. TOR1 imports these VRF INTERNET routes into its local VRF SECURED. Using a local policy, TOR1 colors the routes imported into VRF SECURED with a Color Extended Community named COLFIREWALL for ease of reference. COLFIREWALL has its Color-only Types Field ([I-D.ietf-idr-sr-policy-safi]) set to 1.

Since the routes have a color COLFIREWALL of Color-only type 1, and there is an active color-only SR Policy POLTOR with color COLFIREWALL on TOR1, TOR1 steers the imported routes into SR Policy POLTOR.

Consequently, packets arriving at TOR1 from H12, destined for the Internet, are encapsulated in an outer IPv6 header with a SID list <5f00:0:2:e000::, 5f00:0:6:e000::>. This SID list may be combined into a single uSID container 5f00:0:2:e000:6:e000:: in the IPv6 DA. This SID list directs the encapsulated packets through FW3 and then to BR6.

Each TOR in the DC advertises its VRF SECURED routes with a Color Extended Community that identifies the firewall instance FWX it requires for external communications. The BR steers the routes it receives with this color into the appropriate firewall service.

Each TOR in the DC locally colors the routes imported into VRF SECURED with the color of the local SR Policy that steers egress packets through the same firewall service FWX.

A.2. Cluster Firewall Service

Clustering groups multiple firewalls in a single logical device. The firewall sessions (connections) are synchronized across the devices in a cluster. This ensures seamless failover and uninterrupted traffic flows. It also allows forward and reverse traffic flows of a firewall session to be hashed on different ECMP paths, passing through different firewall cluster members. For example, for a TCP session between H12 and a server on the Internet, the forward flow may be hashed on firewall cluster member FW3, while the reverse flow may be hashed on member FW5 of the same firewall cluster.

All members advertise the same anycast service SIDs to leverage ECMP load-sharing across firewall cluster members. If the firewall cluster is SRv6-unaware, the SRv6 nodes in front of the firewall instances instantiate the same anycast SIDs, directing the traffic to the appropriate firewall instance.

For example, the service leaf nodes connected to firewall cluster members FW3 and FW5 both advertise the anycast locator 5f00:0:24::/48 and instantiate common uA SIDs 5f00:0:24:e000:: and 5f00:0:24:e001:: for the inside and outside firewall interfaces, respectively.

Figure 4 illustrates the network topology with a cluster firewall service with cluster members FW3 and FW5. These firewall services are connected to service leaf (SL) nodes SL2 and SL4, respectively. Each firewall is dual-connected to the SL node via an inside and outside interface.

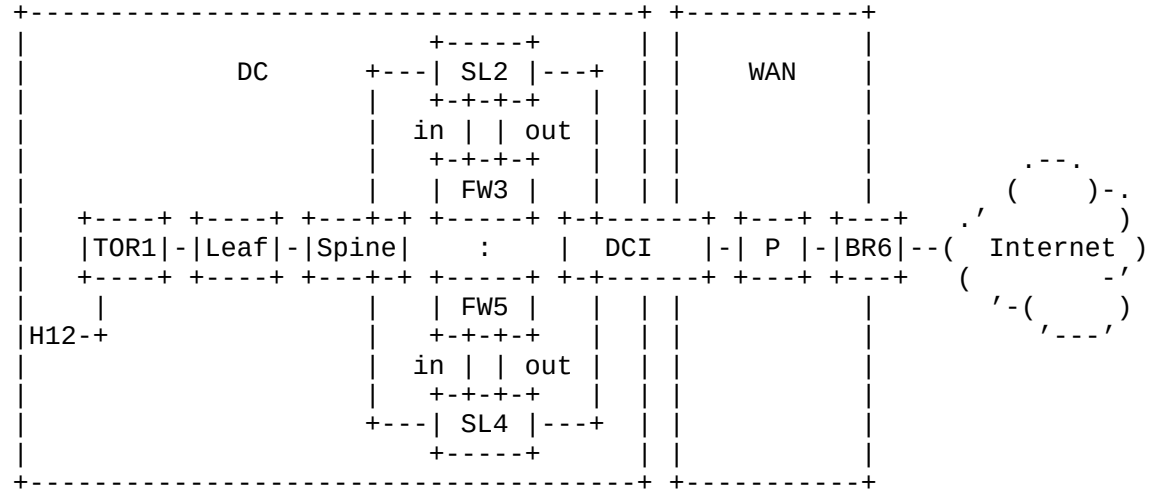

Figure 4: Reference network topology with cluster firewall service

The SL nodes SL2 and SL4 are SRv6-enabled and respectively advertise locator prefixes 5f00:0:2::/48 and 5f00:0:4::/48. In addition, SL2 and SL4 both advertise the same anycast locator prefix 5f00:0:24::/48.

These SL nodes allocate and install a common uA SID (5f00:0:24:e000::) directing packets to the inside interface of the firewall (FW3-IN for FW3, FW5-IN for FW5) and a common uA SID (5f00:0:24:e001::) directing packets to the outside interface of the firewall (FW3-OUT for FW3, FW5-OUT for FW5).

This scenario is similar to the standalone firewall scenario, with the difference being the SID list of the SR Policies to steer the traffic through the firewall service.

SR Policy POLTOR on TOR1, as specified in section A.1, has a SID list <5f00:0:24:e000::>. Due to the anycast locator, this SID list steers the packets to either SL2 or SL4 into the inside interface FW3-IN of FW3 or FW5-IN of FW5, respectively.

SR Policy POLBR on BR6, as specified in section A.1, has a SID list <5f00:0:24:e001::>. This SID list steers the packets to either SL2 or SL4, into the outside interface FW3-OUT of FW3 or FW5-OUT of FW5, respectively.

Appendix B. SRv6 Service Considerations

B.1. SRv6-Unaware Service

If the service is SRv6-unaware ([I-D.ietf-spring-sr-service-programming]), it must be connected to an SRv6-capable entity, a physical or virtual element that can handle SRv6 packets. In section 5, this entity is a physical service leaf node. The requirements for the SRv6 entity depend on the service capabilities.

If the service (e.g., firewall, IPS, IDS, etc.) can process the inner packets without modifying the outer IPv6 header, which may include a Segment Routing Header (SRH), it is sufficient for the connected SRv6 entity to implement uN and uA SRv6 SID behaviors ([RFC9800]) to direct the packets into the service. The service returns the processed packets to the SRv6 entity, where they continue their journey to their destinations.

If the service cannot process encapsulated packets, proxy SRv6 SID behaviors ([I-D.ietf-spring-sr-service-programming]) must be implemented on the connected SRv6 entity. The service processes the decapsulated packet, as the proxy entity provides, and returns the processed packets to the proxy. The proxy restores the packet encapsulation and forwards the packet towards its destination.

B.2. SRv6-Aware Service

The SRv6-aware Service does not depend on an additional connected SRv6 node to execute the SRv6 SID behaviors related to the service. The SRv6-aware service node is reachable via its SRv6 locator and executes the behavior of its local SID matching the outer IPv6 DA of the received packets.

Authors' Addresses

Clarence Filsfils
Cisco Systems
Belgium
Email: cf@cisco.com

Pablo Camarillo Garvia (editor)
Cisco Systems
Spain
Email: pcamaril@cisco.com

Kris Michielsen
Cisco Systems
Belgium
Email: kmichiel@cisco.com

Alexey Gorovoy
Nebius
Netherlands
Email: algorovoy@nebius.com